

REMARKS

This amendment is being filed as a response to the Final Office Action of September 28, 2007. The due date for response extends to December 28, 2007. The only amendment to the claims is to correct a clerical error. Reconsideration is respectfully requested in view of this amendment and remarks. This response is filed **within 2 months**, and therefore, the Applicant respectfully requests expedited consideration in accordance with the rules.

Objections to the Claims

Claim 6 has been objected to because of informalities. Applicant respectfully asserts that such objections are rendered moot in view of the amendment to Claims 6 submitted hereinabove.

Rejections under 35 USC § 103(a)

Claims 1, 6, 8, 9, and 22-24 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Ort et al. (JAXB), and in further view of Betts et al. (U.S. Publication No. 2005/0039166). It should be noted that Patent Application 2005/0039166 from Betts et al. was filed on July 9, 2004, which is after Applicant's filing date of April 1, 2004. Betts et al. Application claims priority date of July 11, 2003 corresponding to Provisional Patent Application 60/486, 755. As a result, Applicants respectfully assert that the Office must rely on the Provisional Application to suggest Applicant's claims, and not on the non-Provisional Application that was filed after Applicants' Application. Not all the teachings from the non-Provisional Application relied upon by the Examiner are found in the Provisional Application, as discussed infra, and these rejections are therefore moot. Applicants respectfully request reconsideration of these rejections in light of the arguments contained herein.

Claim 1 defines accessing a compiled document type definition (DTD) stored in the memory of the device, wherein the compiled DTD comprises executable program code configured to execute on the CPU and cause the CPU to receive the XML document as input, the compiled DTD being a self contained executable program that verifies whether the XML document conforms to a DTD that corresponds to the XML document, the compiled DTD being generated by parsing a DTD document to generate source code, the DTD document containing the DTD corresponding to the XML document, and compiling the source code to generate the compiled DTD.

The Examiner has relied on Paragraph [0052] from Betts to meet the limitation of “compiling the source code to generate the compiled DTD.” However, there is no such teaching in Betts’ Provisional Application that teaches that “the server will try to run the standard functionality implemented by all the smart data classes (e.g. an ‘execute()’ function)” (Page 4, last Paragraph - emphasis added). There is no teaching about “compiling the source code to generate the compiled DTD,” “wherein the compiled DTD comprises executable program code,” as claimed. Running an execute() object associated with a data class is not the same as running executable code that has been compiled, because the execute() function is interpreted when invoked in the object oriented environment of Betts. There is no suggestion in the Provisional Application of executing precompiled code, as claimed. Since there is no teaching in the Provisional Application of compiling source code to generate a compiled DTD, the Examiner’s reliance on Betts is improper.

Claim 1 defines verifying the XML document by running the compiled DTD on the CPU, wherein the compiled DTD receives the XML document as input and generates one of a verified XML output or an error. The Examiner has relied on the following paragraph from Ort to suggest applicant’s limitation:

"This statement highlights an important feature of JAXB: you can have it validate the source data against the associated schema as part of the unmarshalling operation. In this case, the statement asks JAXB to validate the source data against its schema. If the data is found to be invalid (that is, it doesn't conform to the schema) the JAXB implementation can report it and might take further action. JAXB providers have a lot of flexibility here. The JAXB specification mandates that all provider implementations report validation errors when the errors are encountered, but the implementation does not have to stop processing the data. Some provider implementations might stop processing when the first error is found, others might stop even if many errors are found. In other words, it is possible for a JAXB implementation to successfully unmarshal an invalid XML document, and build a Java content tree. However, the result won't be valid. The main requirement is that all JAXB implementations must be able to unmarshal valid documents." (Page 5, first full paragraph - emphasis added)

Ort also teaches that "[a] new Java API called Java Architecture for XML Binding (JAXB) can make it easier to access XML documents from applications written in the Java programming language. However, JAXB, a new JAVA API does not suggest verifying the XML document by running the compiled DTD, as claimed. An API is a programmatic user interface and not a compiled program from source code. Thus, verifying a XML document using JAXB, as in Ort, does not suggest verifying the XML document by running the compiled DTD, as claimed.

Independent Claim 22 is allowable for at least the same reasons as with Claim 1 because it also claims using a compiled DTD to verify the XML document.

Dependent Claim 6 describes that the compiling further includes compiling the source code with a verifier interface to generate the compiled DTD. The Examiner has relied on "the content objects of the classes produced by the binding compiler" to suggest Claim 6 limitations. However, Ort teaches that "the binding compiler generates a set of interfaces and a set of classes that implement the interfaces." Ort uses a binding compiler that is completely different from a program compiler, because Ort's binding compiler generates interfaces and a set of classes to implement the interface, instead of compiled code. Therefore, a binding

compiler does not suggest compiling the source code to generate the compiled DTD, as claimed.

Claim 9 defines that the verifying of the XML document using the compiled DTD further includes generating one of an error, a verified XML document, and the verified XML document with an inserted attribute. The Examiner has relied on the first full paragraph on Page 5 from Ort excerpted above. However, merely suggesting validating the source data against its schema does not teach generating the verified XML document with an inserted attribute, as claimed. More specifically, there is not even a suggestion in Ort of an inserted attribute in the XML document. The mere suggestion of validating source data, as in Ort fails to teach generating ... the verified XML document with an inserted attribute, as claimed. The Office has not articulated reasoning with rational underpinning to support the conclusion of obviousness for the claimed generating the verified XML document with an inserted attribute.

In view of the foregoing, the Office is requested to withdraw the rejection of claims 1, 22, 6 and 9 under §103. Other dependent claims are submitted to be patentable for at least the same reasons the independent claims are believed to be patentable. The Applicants therefore respectfully request reconsideration and allowance of the pending claims. A Notice of Allowance is respectfully requested.

If the Examiner has any questions concerning the present amendment, the Examiner is kindly requested to contact the undersigned at (408) 774-6920. If any other fees are due in connection with filing this amendment, the Commissioner is also authorized to charge Deposit Account No. 50-0805 (Order No. SUNMP365). A duplicate copy of the transmittal is enclosed for this purpose.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP

/Jose M. Nunez/

Jose M. Nunez
Reg. No. 59,979

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
Telephone: (408) 749-6900
Facsimile: (408) 749-6901